

# Day 7 - Introduction to counting reads with featureCounts

## Author:

Rutendo Sigauke ([rutendo.sigauke@colorado.edu](mailto:rutendo.sigauke@colorado.edu))

Updated by Samuel Hunter ([sahu0957@colorado.edu](mailto:sahu0957@colorado.edu)) - 2024

## Introduction:

The featureCounts library is part of Subread (written in C) and RSubread (an R wrapper for Subread), and it is a fast tool optimized for counting reads over features (genes, exons, transcripts .etc). To see the full utility of Subreads/Rsubread, see their documentation below:

Subread: <http://subread.sourceforge.net/>

RSubread: <http://subread.sourceforge.net/SubreadUsersGuide.pdf>

Since counting is compute-intensive, **this is done on the server (AWS)**. Usually, we can request multiple threads which makes the counting run faster. **To save on time and memory, the counts that you will generate are subsetted to chr21 genes only.** After you generate these test counts, we will use the pre-generated full counts data for DESeq2.

**!** If you have not installed Rsubread on the AWS R, do so now ([Rsubread](#)). Installation can be done in the R console. See the [Day7\\_installing\\_Rsubread](#) worksheet for more comments on installing Rsubread.

## Setting up working space:

- Update the srworkshop folder from GitHub (Change the path below to your wherever you cloned the repository)

```
cd /Users/<your_username>/srworkshop  
git pull
```

- Make a working directory for day 7 in your scratch and create subfolders for your error and output files

```
[rutendos@ip-172-31-29-36 ~ $ cd /scratch/Users/rutendos/
[rutendos@ip-172-31-29-36 /scratch/Users/rutendos $ mkdir workshop-day7
[rutendos@ip-172-31-29-36 /scratch/Users/rutendos $ cd workshop-day7
[rutendos@ip-172-31-29-36 /scratch/Users/rutendos/workshop-day7 $ mkdir scripts eofiles
[rutendos@ip-172-31-29-36 /scratch/Users/rutendos/workshop-day7 $ ls
eofiles scripts
[rutendos@ip-172-31-29-36 /scratch/Users/rutendos/workshop-day7 $ pwd
/scratch/Users/rutendos/workshop-day7
```

- Copy the counting script from /Users/<your\_username>/srworkshop to scratch

```
cp /Users/rutendos/srworkshop/projectB/day07/scripts/d7_featureCounts.*  
/scratch/Users/rutendos/workshop-day7/scripts/
```

## Summary of scripts:

There are two scripts we will be using for counting:

1. An R script called **d7\_featureCounts.R**
2. An sbatch script that calls the above R script **d7\_featureCounts.sbatch**

Edit both scripts so that the paths point to your files.

## Counting with featureCounts:

1. Open **d7\_featureCounts.R** with vim. The first command sets your working directory. This is where all your files and figures you will be working on will be saved. Set the working directory to a location of your choice. (e.g. “/scratch/Users/<your\_username>/workshop-day7”)

```
workdir <- ‘/PATH/TO/WORKING/DIRECTORY’  
setwd(workdir)  
getwd()
```

2. The following command loads the **RSubread** package into your R environment. This is the library that has **featureCounts** for counting reads.

```
library(“Rsubread”)
```

3. In order to count reads, we need to give featureCounts a path to our bam files and gene annotation files.

- a. Below we are getting a list that contains full paths to the bam files to be counted.

```
bamdir <- '/scratch/Shares/public/sread2024/data_files/day7/bam'
filelist <- list.files(path=bamdir,
                        pattern="sorted.bam$",
                        full.names=TRUE)
```

- b. The gene annotation file is a gtf format for the human genome (hg38).

```
hg38gtf <- /scratch/Shares/public/sread2024/data_files/day7/annotations/hg38_ucsc_genes_chr21.gtf
```

Below is what the GTF file looks like. This tells featureCounts how to assign each overlapping read to its corresponding feature.

```
chr21 unknown exon 5022493 5022693 . + . gene_id "ICOSLG"; gene_name "ICOSLG"; p_id "P15779"; transcript_id "ICOSLG"; gene_name "ICOSLG"; p_id "P7798"; transcript_version "1"; phase 0; strand +; score 0; gene_id "ICOSLG"; gene_name "ICOSLG"; p_id "P19886"; transcript_id "ICOSLG"; gene_name "ICOSLG"; p_id "P7798"; transcript_version "1"; phase 0; strand +; score 0; gene_id "ICOSLG"; gene_name "ICOSLG"; p_id "P19886"; transcript_id "ICOSLG"; gene_name "ICOSLG"; p_id "P7798"; transcript_version "1"; phase 1; strand +; score 1; gene_id "ICOSLG"; gene_name "ICOSLG"; p_id "P7798"; transcript_id "ICOSLG"; gene_name "ICOSLG"; p_id "P19886"; transcript_version "1"; phase 1; strand +; score 1; gene_id "ICOSLG"; gene_name "ICOSLG"; p_id "P15779"; transcript_id "ICOSLG"; gene_name "ICOSLG"; p_id "P15779"; transcript_version "1"; phase 0; strand +; score 0;
```

**NB:** Take a look at the GTF file structure. Note all the different features represented for each feature. Also, you will see that the file has several columns, with the **first** column is the chromosome ID, the **second** column is the name of the source from which the feature was derived (eg. RefSeq, Ensembl, UCSC or HAVANA). The **third** column is the label for the feature (e.g. exon, CDS, start\_codon). This field is used by featureCounts to determine the features to count reads over. The **fourth** and the **fifth** columns are start and end coordinates respectively. The **sixth** column is the score of the feature, the **seventh** is the strand, the **eighth** is the phase for CDS features (If phase=0, the codon begins at the first base of CDS nucleotide; if phase=1 the codon begins at the second base of CDS nucleotide; if phase=2 the codon begins at the third base of CDS nucleotide.). Lastly, the **ninth** column contains additional feature annotations.

4. The featureCounts() command is shown below taking in the paths to bam files and gene annotations. The command also allows the user to specify the feature (GTF.featureType) to count over. Since this is RNA-seq data we are counting over exons. Additionally, we can set the name to assign the features (GTF.attrType) as “gene\_id”. Please check the documentation for the **featureCounts()** command to get more information on all the flags.

```
fc <- featureCounts(files=filelist,
                      annot.ext=hg38gtf,
                      isGTFAnnotationFile=TRUE,
                      GTF.featureType="exon",
                      GTF.attrType="gene_id",
                      useMetaFeatures=TRUE,
                      allowMultiOverlap=TRUE,
                      largestOverlap=TRUE,
                      countMultiMappingReads=TRUE,
                      isPairedEnd=TRUE,
                      strandSpecific=2,
                      nthreads=4) #when you move to a bigger machine change to 8
```

5. We can also set the output folder. Note that this folder is based on the workdir from above. We can also create this new folder within R for where the counts will be saved.

```
outdir <- paste(workdir,'/', 'counts', '/', sep='') ##naming our outdir
dir.create(outdir) ###creating the directory
```

6. Once the counts are obtained, the outputs can be saved as tab-separated files.
  - a. The GenID, gene length, and counts are saved to a single file here:

```
write.table(x=data.frame(fc$annotation[,c("GeneID", "Length")],
                         fc$counts, stringsAsFactors=FALSE),
            paste0(outdir, fileroot, file="_featureCounts_gene_rnaseq.txt"),
            quote=FALSE, sep="\t",
            row.names=FALSE)
```

- b. Separate counts data and summary statistics.

```
write.csv(fc$counts, paste(outdir, fileroot,".coverage.csv", sep=""))
write.csv(fc$stat, paste(outdir, fileroot,".stat.csv", sep=""))
write.csv(fc$annotation, paste(outdir, fileroot,".annotation.csv", sep=""))
write.csv(fc$targets, paste(outdir, fileroot,".targets.csv", sep=""))
```

7. The summary of counts will be in the output counts folder. There are five different files:

- **featureCounts\_gene\_rnaseq.txt** : GeneID, Length, Counts
- **coverage.csv** : Counts
- **.stat.csv** : Coverage Statistics
- **.annotation.csv** : GeneID, Chromosome, Start, End, Strand, Length
- **.targets.csv** : file names for input `bam` files

8. Edit the sbatch script including the **SBATCH headers** and path to the **d7\_featureCounts.R** script. Now we can run featureCounts by submitting the sbatch script!

```
#!/bin/bash

#SBATCH --job-name=<NAME OF JOB>                                # Job name
#SBATCH --mail-type=ALL                                         # Mail events (NONE, BEGIN, END, FAIL, ALL)
#SBATCH --mail-user=<YOUR E-MAIL ADDRESS>                         # Where to send mail
#SBATCH --nodes=1                                                 # Number of cores job will run on
#SBATCH --ntasks=4                                                # Number of CPU (processors, tasks)
#SBATCH --time=1:00:00                                             # Time limit hrs:min:sec
#SBATCH --partition compute                                       # Job queue
#SBATCH --mem=4gb                                                 # Memory limit

#SBATCH --output=/YOUR/EOFILES/PATH/%x_%j.out
#SBATCH --error=/YOUR/EOFILES/PATH/%x_%j.err

##### SET VARIABLES #####
FEATURECOUNTS=/PATH/TO/YOUR/d7_featureCounts.R

#####
##### PRINT JOB INFO #####
printf "Sample ID: $ROOTNAME"
printf "\nDirectory: $PROJECT"
printf "\nRun on: $(hostname)"
printf "\nRun from: $(pwd)"
printf "\nScript: $0\n"
date

printf "\nYou've requested $SLURM_CPUS_ON_NODE core(s).\n"
#####

Rscript $FEATURECOUNTS
```

9. Open and explore each of the files in the terminal (with **head** or **more**).

- **chr21\_Ethan\_Eric\_featureCounts\_gene\_rnaseq.txt**

This is used as input to differential gene expression analysis packages such as DESeq2.

```
[rutendos@ip-172-31-18-92 counts]$ head chr21_Ethan_featureCounts_gene_rnaseq.txt
GeneID  Length    chr21Eric.repA.sorted.bam    chr21Ethan.repA.sorted.bam
ICOSLG  6757     1253     2266
C21orf33 3334     1333     2833
PWP2    6520     490      1438
LINC00313 1158     0        0
LINC00319 6004     0        0
SIK1    9404     426      1083
CBS     5456     114      135
U2AF1   2040     1817     3023
CRYAA   2288     0        2
```

#### - [chr21\\_Ethan\\_Eric.coverage.csv](#)

```
[rutendos@ip-172-31-18-92 counts]$ head chr21_Ethan_Eric.coverage.csv
"", "chr21Eric.repA.sorted.bam", "chr21Ethan.repA.sorted.bam"
"ICOSLG", 1253, 2266
"C21orf33", 1333, 2833
"PWP2", 490, 1438
"LINC00313", 0, 0
"LINC00319", 0, 0
"SIK1", 426, 1083
"CBS", 114, 135
"U2AF1", 1817, 3023
"CRYAA", 0, 2
```

#### - [chr21\\_Ethan\\_Eric.stat.csv](#)

```
[rutendos@ip-172-31-18-92 counts]$ head chr21_Ethan.stat.csv
"", "Status", "chr21Eric.repA.sorted.bam", "chr21Ethan.repA.sorted.bam"
"1", "Assigned", 104000, 202339
"2", "Unassigned_Unmapped", 679, 578
"3", "Unassigned_Read_Type", 0, 0
"4", "Unassigned_Singleton", 0, 0
"5", "Unassigned_MappingQuality", 0, 0
"6", "Unassigned_Chimera", 0, 0
"7", "Unassigned_FragmentLength", 0, 0
"8", "Unassigned_Duplicate", 0, 0
"9", "Unassigned_MultiMapping", 0, 0
```

## - chr21\_Ethan\_Eric.annotation.csv

## - chr21\_Ethan\_Eric.targets.csv

```
[rutendos@ip-172-31-18-92 counts]$ cat chr21_Ethan_Eric.targets.csv
 "", "x"
 "1", "chr21Eric.repA.sorted.bam"
 "2", "chr21Ethan.repA.sorted.bam"
```